Prosessrapport

Studentevalueringssystem

1 Forord

1.2 Forord

Denne prosessrapporten forteller om gruppens samarbeid og metoder benyttet for hovedprosjektet ved Høgskolen i Oslo avdeling for ingeniørutdanning, anvendt datalinjen, vårsemesteret 2010. Medlemmer bak S2S "student to student" er John Abrahamsen og Sara Khelifi.

Rapporten er beregnet for sensor, veileder, oppdragsgiver og andre som har interesse av å sette seg inn i utfordringer, løsninger på disse og arbeidsmetoder i prosjektet.

Det blir i rapporten gjort rede for samsvaret mellom sluttproduktet i forhold til kravspesifikasjonen.

Det kreves ingen spesiell datakompetanse for å sette seg inn i denne rapporten. Dokumentet er optimalisert for papirutskrift.

Innholdsfortegnelse

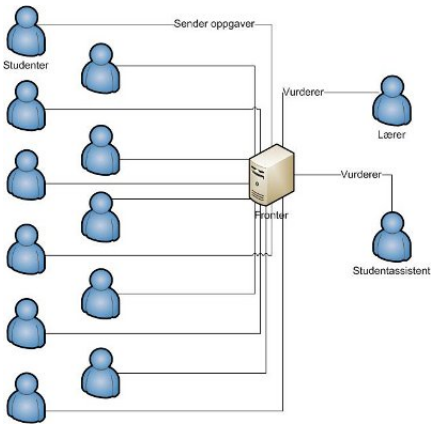
2 Innledning

2.1 Om bedriften

Høgskolen i Oslo er landets største statlige høgskole med 12 000 studenter og 1250 tilsatte. HiO har sju avdelinger med 33 bachelor studier og 18 masterstudier. En så stor utdanningsinstitusjon med så mange studenter krever mye av lærere og studentassistenter så et slikt evaluerings system vil definitivt være med å lette de ansattes hverdag på en positiv måte. Foreløpig vil systemet være fokusert på datastudenter og vil ikke omfatte alle avdelinger i Høgskolen i Oslo.

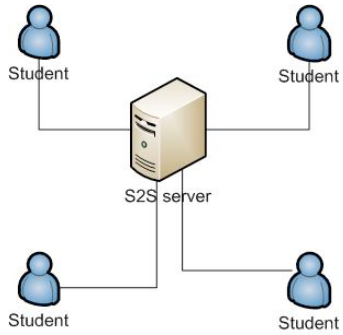
2.2 Dagens situasjon

Som situasjonen er i dag så er det veldig ressurskrevende å få gjort vurdering av elevarbeid. Studenter benytter seg av skolens student portal, Fronter, for å levere inn arbeid for vurdering. Ingenting er automatisert, så det er lærere og studentassistenter som får oppgaven med å gå inn på Fronter å vurdere alt arbeid gjort av studenter. Karakterer eller bestått/ikke bestått settes inn sammen med kommentarer på Fronter så studenter kan lese tilbakemeldingen. Fronter er et bra system, men det har ingen funksjon for å muliggjøre "peer review" som vi beskriver det. Med tanke på hvor mange studenter som studerer på HiO kan man tenke seg til at det er mye ressurser som kan frigjøres ved å kunne designe ett slikt system.



Figur 1 Dagens Løsning illustrert.

Skolen har selv ytret ønske om å kunne lette arbeidsmengden ved å la studenter være delaktig i vurdering av hverandres arbeid.



Figur 2 Ønsket løsning.

Ingeniøravdelingen har utviklet ett system, kalt Mefisto, som har blitt brukt i mindre grad. Etter samtaler med samtlige utviklere av Mefisto kommer det frem at systemet var for vanskelig og tidskrevende å benytte seg av når det ble lansert. Bruken av systemet ble for teknisk avansert for andre enn de som hadde vært med å utviklet det. Systemet har ikke noe grensesnitt og måtte kodes om for hver gang det skulle brukes til noe. Forøvrig var peer review bare en av funksjonene i Mefisto, og ikke den som var "mest utviklet" av de alle. En av ønskene til oppdragsgiveren er at vi skal vurdere om Mefisto kan videreutvikles eller brukes som utgangspunkt for utvikling av en ny applikasjon. Mefisto ble bare brukt i IU-avdelingen på HiO.

2.3 Mål

De sentrale mål for systemet er hovedsakelig det som kreves for at systemet skal fungere riktig i forhold til miljøet og arbeidsplassen.

Systemet bør:

- gjøre det mulig for studenter å vurdere hverandres arbeid.
- gi vurderinger som er troverdige og kan benyttes som at de er riktige, dvs. at det ikke skal være mulighet for fusk.
- være trygt nok til at man ikke kan bryte seg inn å legge inn falske resultater.
- holde orden på hvem som har tilgang til å rette oppgaver og at den som har rettet en oppgave ikke har rett til å rette samme oppgave igjen.
- opprettholde personvernet til den enkelte.
- være 100% nettbasert.
- være åpen kildekode og fritt tilgjengelig.

Funksjonalitet som kan implementeres er:

- varsel på email/SMS når du skal evaluere en oppgave, eller har mottat resultater.

2.4 Rammebetingelser

Gruppen står fritt til å velge hvilket språk vi vil benytte for å programmere applikasjonen. Vi står også fritt til å velge hvilken grad applikasjonen skal være ferdig utviklet. I utgangspunktet tenker vi i arbeidsgruppen at vi må modellere/beskrive systemet 100% og minst lage en proof-of-concept applikasjon. Gruppen planlegger å bruke PHP som programmeringsspråk og MySQL som database verktøy, ettersom det er disse verktøyene vi har mest erfaring med når det gjelder programmering av web-løsninger. Uansett kan man jo tilpasse applikasjonen relativt enkelt til å benytte annen database programvare.

3 Planlegging

3.1 Generelt

Gruppesammensetning var klar en god stund før prosjektoppgaven, noe som gjorde det enklere å begynne med prosjekt søket. Gruppen har samarbeidet tidligere og vet hvor hver av dem står i forhold til den enkeltes kompetanse nivå. Planleggingen av prosjekt søket startet så fort vi fant et prosjekt som passet en gruppe på to. Vi ble raskt interessert i peer review og tok kontakt med oppdragsgiveren. Dermed fikk vi ansvaret for å utvikle et student evalueringsystem.
Vi opprettet etter hvert en blogg side fordi vi fant ut at det er mye enklere og legge ut innlegg som en slags dagbok. Det vil gjøre det lettere for veileder og kontaktperson å følge med.

Til å begynne med var det meget viktig og finne ut definisjonen av peer review før man kunne gå videre med å sette opp planlegging av prosjektet. Deretter ble det satt opp flere møter med oppdragsgiveren for å avklare kravene under planleggingsfasen. Dette ble gjort med god kvalitet og minimaliserte behovet for avklaringer senere i prosjektet. Det var hyppige møter med veileder som skulle holde orden på at vi kom oss videre og at vi leverte resultater for hver uke. Det var viktig å finne ut hva det tidligere systemet sto for og hvorfor det ikke er i bruk i dag. Det var også lurt å undersøke om hvilke muligheter det fantes der ute av kommersielle systemer. Dette vil hjelpe oss vurdere om hva som lønner seg best for HiO. Det er begrenset med ressurser derfor var det viktig at systemet skulle være webbasert. Vi satte også fokus på å utvikle ett godt brukergrensesnitt med god brukervennlighet, noe som oppdragsgiveren ytret ønske om.

3.2 Fremdriftsplan og arbeidsplan

Fremdriftsplanen ble satt opp tidlig i prosjektet. Sammen med denne ble det også utarbeidet en overordnet arbeidsplan. Begge disse dokumentene har blitt brukt aktivt gjennom hele prosjektgjennomføringen, de finner man under styringsdokumenter i slutt rapporten.

4 Utviklingsprosessen

4.1 Fasene i prosjektet

4.1.1 Forprosjekt

Vi har vært heldige om samarbeidet med oppdragsgiver, det ble aldri satt noen krav om hvordan vi skulle jobbe og hva slags prosjektstyringsmetode som skulle brukes. Dette ga oss friheten til å gjøre som vi ville i forhold til prosjekt styring. Gruppen besto kun av to medlemmer dermed var det naturlig å følge en iterativ utvikling. Årsaken var at vi ikke trengte noen spesiell styring ettersom oppgavene var relativt lette å fordele. Vi har kommunisert hele veien og samarbeidet tett med faste møter på skolen og jobbet jevnt med å holde hverandre oppdatert.

I starten av prosjektet var det veldig viktig å sette seg inn i dagens situasjon og hvordan "Mefisto" fungerte tidligere derfor var det gjort grundig forarbeid om det i forprosjektet. Det ble gjort et møte med Mark Burgess en tidligere HiO professor, grunder av CFEngine, og utvikler av Mefisto . Møtet var meget informativt og en del ukjent fakta kom frem.

Oppdragsgiveren hadde noen funksjonelle krav og et ønske om å undersøke og trekke konklusjoner om hva som lønner seg best i forhold til videreutvikle Mefisto, vurdere et kommersielt system eller skape noe nytt. Så en liten evaluering måtte man komme med, ellers sto vi fritt til å velge hvordan vi skulle løse dette teknisk.

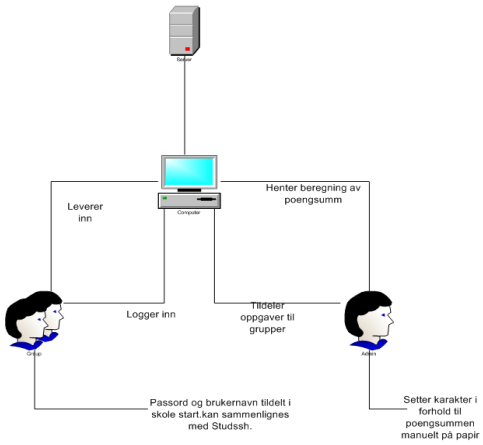
Gruppen var enig om at miljø for testingen driftes på en virtuell Debian Linux maskin fra HiO.
Teknologien som skal brukes er PHP Hypertext Preprocessor, database Mysql, med en blanding av programmeringsspråk som PHP, SQL og Javascript.

Etter samtaler med samtlige lærere så var vi alle enige om at et brukervennlig grensesnitt er et fokus i og med at det forrige systemet ikke hadde noe som helst brukergrensesnitt. Dette er også et noe av det vi har som mål innenfor de rammebetingelser vi har satt opp i kravspesifikasjonen.

4.1.2 Mefisto

Mefisto ble utviklet for ti år siden av Mark Burgess daværende foreleser ved IU avdelingen på Høgskolen i Oslo. En innebygget del av dette systemet var Peer Review der studenter la inn poeng for hver oppgave og systemet tok da en full poengberegning. Programmet ble utviklet med tanke på at det skulle lette hans arbeid for retting av oppgaver. Studenter skulle i grupper gå inn og signere en obligatorisk oppgave, de fikk da et passord og brukernavn og de måtte logge seg inn alle samtidig innen et tidspunkt. Dette fungerte i en tid hvor Fronter ikke eksisterte dermed var det genialt men ganske knotete. Det ble senere utviklet videre av Hårek Haugerud og brukt til flervalgs tester (Multiple choice Test) og dette fungerte helt greit. Problemet

var at det ble kun brukt av få, de som forstår seg på PHP programmering, det hadde ikke noe brukergrensesnitt.
Mefisto har ikke oppdatert seg siden og ingen har tatt arbeidet videre. Systemet er for komplisert uten noe som helst brukergrensesnitt noe som gjør det vanskelig å jobbe med videre.



Figur 3 Peer Review Løsning

4.1.3 Peer Review

En fordel med peer review ordningen er at elevene ikke bare skriver egne oppgaver, men lærer også mye av å rette andres arbeid. Studenten for innsyn i hverandres besvarelser og lar dem reflektere over mulige måter å bli bedre på. Dette vil gi studenten en betydelig ”karakter belønning” ved å se andres perspektiver og lære mer i motsetning til den tradisjonelle måten. Denne måten å arbeide på oppfordrer studenter til å revurdere sin egen ytelse basert på hva de har sett. Dette er et viktig fundament for Peer review mener Mark Burgess fra rapporten Security and online student evaluation with large classes (Burgess.M.2003[1]).

I følge rapporten av Mark Burgess så var studentene i Høgskolen i Oslo fornøyd med peer review, de var fornøyd med å evaluere hverandres arbeid å bli evaluert av studenter, men noen var skeptiske for å bli vurdert av studenter og viste usikkerheter til å la andre studenter evaluere. Mange tenkte at ikke alle studenter ville være troverdige under evalueringen slik en sensor ville ha vært. Det som undervurderes er at ikke alle studenter forstår at de har opparbeidet seg bedre karakterer. Ved å se på hvordan andre har løst oppgaver og som nevnt ovenfor revurdere sitt ytelse basert på hva de har sett.
Dette er også noe av grunnlaget vi har basert utviklingen av s2s på, nemlig å lette foreleserens arbeid, men studenten vil også komme bedre ut av dette i forhold til kunnskaps tilegnelse og opparbeidelse av karakter ,altså en vinn - vinn situasjon.

4.2 Kompetansetilegning

På Høgskolen i Oslo, avdeling for ingeniører, har studentene hatt fag som MySQL og PHP igjennom faget ”Databaser” og ”Webprogrammering” som kjøres i første og andre året.
Disse fagene har hele veien vært meget relevante både til eksamen og andre prosjekt oppgaver derfor har vi hele veien tatt til betraktning disse fagene og prioritert de høyt. Noe som nå kommer til godt bruk med på hovedprosjektet.

Oppdragsgiver framsatte ikke krav om hvilken teknologi vi skulle bruke, men det var ingen tvil om at dette Mysql og Php skulle bli tatt i bruk. Dette er verktøyer som i dagens utviklings miljø blir brukt mer og mer sammen i utvikling av databasesystemer. Gruppen satte seg som mål å lære seg bruk av disse på et relativt høyt nivå gjennom utviklingen av systemet, slik at vi tilslutt kunne presentere et kodemessig solid system.

Da gruppen kun bestod av to medlemmer, ble det gjort slik at den ene som hadde bedre programmerings kunnskaper var den som sto for programmerings biten, mens det andre gruppe

medlemmet fokuserte på modellering , design og rapportering. Det ble brukt phpmyadmin for oppsett av databasen. Dette gjorde at systemet ble mer oversiktlig og lettere og utvikle underveis, dessuten nyttig for fremtidige oppdatering/ utvikling.

4.3 Programmering

Før vi startet med programmeringen av applikasjonen diskuterte vi og modellerte systemet slik vi ville at det skulle se ut, med tanke på funksjonalitet og utforming. Vi brukte ett program som heter Violet UML for å lage use cases og vi lagde klassediagram i ArgoUML. Når vi hadde fått en oversikt over hvilke funksjoner som skulle være med startet vi med å programmere.

I og med at vi var to på gruppen delte vi oss og den ene fikk ansvar for å programmere og den andre fikk ansvar for dokumentasjon av prosessen. Dette var en grei inndelig og vi fikk jobbet bra. Selvfølgelig var begge delaktig i utviklingen, ofte når det kom til beslutninger som måtte tas og spørsmål om grensesnitt.

I utgangspunktet hadde vi tenkt at applikasjonen skulle bli en proof-of-concept applikasjon, men etter hvert fikk vi forespørsel fra oppdragsgiver om vi kunne noe som kunne brukes i praksis på skolen. Det ble selvfølgelig en mye større arbeidsmengde, men vi var sikre på at vi skulle klare å komme i mål med applikasjonen og resten av prosjektet.

Utviklingen ble delt inn i iterasjoner, hvor vi stadig økte og endret funksjonalitet. Det ble hele tiden gjort evolusjoner av eksisterende funksjoner etter hvert som vi programmerte. Det er lettere å ta små steg om gangen enn å sitte med ett stort uferdig produkt som ikke kan brukes til noe, hvertfall når det er en så liten gruppe som jobber. I og med at det ikke er noe kommersielt produkt som er tilgjengelig har alt som er blitt laget vært basert på egne tanker og ideer. Det er vanskeligere å gjøre det enn når man har eksisterende konkurrenter som settes som mål i forhold til funksjoner, utforming osv.

4.4 Testing

Å levere et feilfritt system til oppdragsgiver ligger ganske nært til umulig, det vil alltid dukke opp noe. Derfor er det viktig å dokumentere hva som er testet og hvordan, for å lette arbeidet for de som skal vedlikeholde systemet og finne eventuelle feil.

Det første som ble testet, var databasen og sql spørringer. Dette testet vi selv uten å bruke testpersoner. Deretter var det GUI som ble testet. Vi utviklet en papir prototype for å først få et bilde av hvordan vi så for oss at grensesnittet skulle være, deretter ble det testet på vanlige brukere. Vi fikk veldig god tilbakemelding, og selv om det ble lagt stor vekt på programmerings biten av databasen og ramme verket så har det ikke vært så mye tid til testing av GUI. Vi tok godt tak i de få tilbake meldingene vi hadde fått og forbedret så godt vi kunne.

Testingen var delt i tre forskjellige deler, Administrator delen, student bruker delen og lærer delen.
Gruppen valgte en iterativ uvikling av systemet. Dette innebar at systemet ble ferdigstilt modul for modul. En ny modul ble aldri påbegynt før den sist utviklede modulen viste seg å fungere som forventet.

I hovedprosjekt blir det som oftest fokusert på funksjonell testing, og vårt prosjekt var intet unntak. Vi startet med såkalt ”glass-box” testing, hvor man bruker kjennskap til koden for å gå igjennom systemet å kjøre tester. Senere når det ble satt mer fokus på at modulene gjorde det de skulle, brukte vi ”black-box” testing, altså matet systemet med testdata for å teste hvilken tilbakemelding det ga. Her trengs ikke kjennskap til koden.

4.5 Utfordringer i prosjektperioden

4.5.1 Utviklingsmiljø

Det har oppstått en del faglige og tekniske utfordringer underveis men alt har latt seg løse. Vi føler som gruppe at vi sitter igjen med mye kunnskaper rundt dette prosjektet en det vi gjorde til å begynne med.

Før vi startet utviklingen jobbet vi med å få skaffe verktøy for å kunne klare jobben så raskt og smertefritt som mulig. For å lage en web-server som var tilgjengelig overalt valgte vi å benytte oss av en virtuell maskin fra skolen. Denne ble satt opp med Ubuntu. Når vi fikk den fra skolen med brukernavn og passord var det en Ubuntu-klient. Vi satt i gang med å konfigurere den etter vårt behov. Vi fjernet det grafiske grensesnittet og alle de unødvendige tjenestene som kjørtes, satt opp en brannmur osv. Når alt var satt opp slik vi ville så installerte vi Apache med php-modul, MySQL og phpmyadmin. Vi konfigurerte så tilgangsrettigheter på Apache og databasen. Til slutt satt vi med en VM som fungerte veldig bra som server. Vi satt også opp en last overvåker (Munin).

På nettsiden benyttet vi oss av en rekke programmer for å gjøre utviklingen enklere. Først og fremst brukte vi Dreamweaver for å designe det grafiske grensesnittet til applikasjonen, samt å ha en plass å få en overordnet bilde over alle moduler av applikasjonen. Dreamweaver fungerer veldig bra til å designe grafiske elementer, det vil si CSS og HTML, men gir lite til ingen hjelp når man skal programmere PHP. For å lage ER-diagrammer brukte vi MySQLWorkbench, som er en videreutvikling av MySQL Front. Denne applikasjonen er veldig hendig ettersom den kan brukes til å administrere databaser og veldig mye annet. Vi brukte den til å lage en spørring som kjørtes for å opprette databasen. Den er derimot ikke så veldig bra til å lage SELECT,INSERT og DELETE spørringer. For å gjøre det brukte vi ett program som heter RazorSQL. Dette var til stor hjelp for å lage en del spørringer, de mest avanserte var noe man måtte gjøre ”frihånd”, men det var veldig kjekt med ett program som fort løste de enkle spørringene som å hente en oppgave.

Gruppen fikk tildelt et brukernavn og passord for vm som blir brukt i ingeniøravdelingen, dette ga oss et område å serve domenet vårt på. Vi brukte et gratis domenet fra no-ip.org.

4.5.2 Optimalisering av GUI

GUI er viktig. Det er faktisk det eneste brukerne ser av applikasjonen. Derfor er det viktig at både personer og nettlesere forstår applikasjonen. Vi har jobbet en del med universell utforming i kurset vårt. Derfor bør vi få vist at vi kan utforme med tanke på at det ikke skal være vanskelig får mennesker og forstå. Vi skal utføre tester av grensesnittet vårt og gjøre forbedringer etter de innspillene vi får fra testpersoner.

En annen ting er at applikasjonen bør helst fungere i de nettleserne som blir mest brukt. Vi tenker da på Internet Explorer, Firefox, Opera, Safari, Google Chrome. Som nevnt må funksjonaliteten være den samme, men den grafiske fremstillingen bør også være så å si den samme uten for mye avvik fra slik det er tiltenkt å fremstå. En god start for å oppnå dette er å få nettsiden godkjent av W3C XHTML validatoren. Vi kommer til å utføre tester med de forskjellige nettleserne samtidig som vi gjør tester av brukergrensesnittet på mennesker.

5 Kravspesifikasjonen

5.1 Generelt

Da kravspesifikasjonen skulle utvikles, kontaktet vi oppdragsgiver for å få høre deres ønsker og eventuelle krav de måtte ha til systemet. Dette, i tillegg til våre tanker og ideer, dannet kravspesifikasjonen.
Kravspesifikasjonen skal betraktes som en kontrakt mellom prosjektgruppen og oppdragsgiver, men vi var enige om at den ikke skulle stå i veien om vi fikk nye og bedre ideer etter hvert.

5.2 Endringer i Kravspesifikasjonen

Første versjon av kravspesifikasjonen ble skrevet i forprosjektfasen. Denne var basert på ønsker vi fikk fra oppdragsgiver, i tillegg til den funksjonalitet gruppen så for seg systemet hadde ved prosjektslutt. Da vi kom ut i planleggingsfasen, ble det tydeliggjort at kravspesifikasjonen var en smule vag på noen områder og enkelte ønsker fra oppdragsgiver var litt utydelige. Gruppen bestemte seg for at kravspesifikasjonen måtte utbedres og det ble avtalt et møte med veileder der det ble utarbeidet en bedre forklaring på hva en kravspesifikasjon skal innholde.

Første versjon av kravspesifikasjonen gikk mest ut på å definere minstekrav til systemets funksjonalitet og tekniske krav. I tillegg til dette, inneholdt andre versjon mer spesifikk funksjonalitet, mer krav til hva systemet skulle kunne levere. Andre versjon var også utvidet en del, siden veileder rettet den første versjonen og fant uklarheter. Gruppen var fornøyd med denne versjonen en stund, men da vårt bilde av systemet etter hvert ble klarere og klarere fant vi ut at det vil bli en del avvik i systemet i forhold til kravspesifikasjonen men dette er nødvendigvis ikke noe negativt. Det var nok av klare krav til hva systemets forskjellige komponenter skulle ha av funksjonalitet. Dette ble den endelige versjonen av kravspesifikasjonen. Vi følte at denne versjonen helt og holdent representerte systemet vi skulle utvikle.

5.3 Kravspesifikasjonens rolle under design og implementasjon

Vi fikk etter hvert en veldig oversiktelig kravspesifikasjon, som vi kunne forholde oss til under utviklingen av systemet.
Første versjon av GUI(papirprototypen) ble utarbeidet på bakgrunn av kravene satt i kravspesifikasjonen. Da et webbasert GUI skulle utvikles, måtte det gjøres noen forandringer. Disse forandringene må sies å være til de bedre for systemet, og til tross forandringer har gruppen aldri gått mot kravspesifikasjonen under utviklingen.

I kravspesifikasjonen er det tydeliggjort hvilken funksjonalitet oppdragsgiver og gruppen har ønsket for systemet, og disse har også hatt stor prioritet under

utviklingen. Det har hele tiden blitt arbeidet mot denne under implementasjon av kode til systemet. De mest grunnleggende kravene(som logg inn, registrering, sletting) ble implementert først, deretter det vi fant mest viktig. Dette ble gjort i tilfelle at uansett om gruppen hadde tatt seg vann over hodet tidsmessig, ville oppdragsgiver få den mest grunnleggende funksjonalitet på systemet.

Kravspesifikasjonen har under hele prosjektperioden vært en fin huskeliste å forholde seg til. Har det vært spørsmål angående funksjonalitet har vi alltid dobbeltsjekket med kravspesifikasjonen. Uten denne har det villet vært svært vanskelig å få fram det resultatet vi har i dag.

5.4 Kravspesifikasjonen i dag

Alle punktene i kravspesifikasjonen er ikke oppfylt grunnet tidsmangel og prioriteringer av de forskjellige funksjonaliteter.Vi regner med at det blir noe lavere score på enkelte punkter under evalueringen pga av dette. Tross alt er det ikke alt for mye kode som skal til, for å fullføre smutthullene siden databasen er klargjort for alt i kravspesifikasjonen.

5.5 Avvik

Det er en del avvik fra kravspesifikasjonen som nevnt ovenfor ikke ble oppfylt enten i systemet elli GUI.

5.5.1 Avvik i evalueringssystemet

1. Ha avansert beskyttelse mot utryggheter som sql injections.

2. Motta varsel på e-post når en oblig er rettet.

5.5.2 Avvik i Administrasjonsdelen

1. Import/eksport av elevdata fra student databsen.
2. Ingen statistikk del.

5.5.3 Avvik i brukergrensesnitt

1. Utskrift ikke lagt til
2. Oversettelse til engelsk for utlandske studenter ikke lagt til

Ikke all funksjonalitet som vi har satt som krav til systemet har blitt løst. Dette kan anses som avvik men vi ser på det som forbedringsmuligheter i fremtiden. De fleste avvik er heller ikke så store, men små endringer som ikke tar så lang tid å sette opp.

6 Oppsummering

6.1 Produktet

Vi har utviklet et produkt som har oversteget de mål vi har satt oss til å begynne med, noe vi er meget fornøyd med . I starten så trodde vi aldri at vi skulle få til et fungerende system men kanskje et proof of concept og at rapporten skulle ta en mer faglig vinkling. Dette viste seg til å snu helt om, da vi under prosessen så at det var fullt mulig å lage ett produkt som fungerer.

Per dags dato så fungerer det meste av primær funksjonene helt optimalt og har oversteget de krav som ble satt opp i kravspesifikasjonen. Dette systemet er det mest lønnsomme å bruke enn å finne et kommersielt system eller å bruke det ti år gamle Mefisto. Systemet Mefisto har ikke oppdatert seg siden og har ikke noe grensesnitt. Dette kan man se nærmere på under punkt 4.1.1-4.1.3.

Vi er utrolig fornøyd med å ha fått til backend og et brukergrensesnitt. S2S kan virke for en utenforstående som et enkelt system som kunne ha tatt kortere tid til å bygge opp men det har vært ganske utfordrende til tider og sette opp database og spørringer. Det å finne enklere løsninger til funksjonalitet og problemstillinger har ikke alltid gått så lett, men det er vel noe

som er ganske vanlig i et prosjekt arbeid. Vi har måttet sette oss inn i nye programmeringsspråk på et dypere nivå enn det vi har gjennomgått i tidligere fag. Å finne nye metodikker og teknologier for å implementere underveis i prosessen har vært ganske stort og omfattende. Og dette krever mye tid i tilegg til å faktisk få det til.

Brukergrensesnittet har et gjennomgående enkelt og intuitivt design, men hovedfokuset var å løse de oppgaver som er ment man skal gjøre i et student evaluerings system. Med tanke på tidspresset og fordelingen av både rapport skriving og utviklingen av produktet så har prioriteringene for et brukervennlig og dynamisk brukergrensesnitt fått en del avvik.

Oppdragsgiver er veldig fornøyd med utfallet av prosjektet. Han ble ble veldig imponert over at vi var i stand til å lage ett produkt som holdt den standarden den gjorde. Arbeidsgiveren har gitt lite feedback på hva han vil ha i applikasjonen, vi har kun hatt noen få møter med han, men vi har likevel klart å løse hans krav. Det meste har gått mellom veileder. Arbeidsgiveren ble også fornøyd med det grafisk grensesnittet vi hadde utarbeidet. Han ga uttrykk for at denne applikasjonen kunne brukes på en klasse allerede til neste semester i samarbeid med en foreleser for å teste ut systemet. Han anbefalte at vi fortsetter med utviklingen av prosjektet våres selv etter at vi var ferdige, for dette var et produkt som Høgskolen i Oslo og andre kunne være interreserte i.

6.2 Prosessen

Samarbeidet gjennom hele prosjekt perioden har gått veldig bra. Gruppen har samarbeidet sammen tidligere opp gjennom de tre årene på høgskolen og kommer godt overens. Gruppen vet hvor man har hverandre både på kompetanse nivå og kapasitets nivå. Hardt arbeid og fokus er mange av de personlige mål som gruppen har til felles.

Dette prosjektet skiller seg ut fra andre prosjekter fordi vi har klart å utvikle kode fra bunn av og ikke satt fokus på å forstå tidligere koder eller å implementere ny kode.

Vårt system har også et annet formål en å lage en nettbutikk, ett regnskapsystem eller ett nettsted som de fleste andre prosjekter. Student to student er ett etterspurt student evalueringssystem som er genialt, det vil gi lærere rom og tid til mye annet å være en fordel for studenter ved at de lærer av hverandres oppgaver. Dessverre så er det ingen som har hatt ressurser eller tid til å utvikle det videre eller på nytt, og dette ser vi på som ett kupp som har fått anledning til å utvikle dette systemet. Vi føler oss ganske heldige som har fått i oppgave av HiO å utvikle dette systemet.

Vi i gruppen føler vi har vokst veldig på den erfaringen vi har opparbeidet oss under prosjektgjennomføringen og de problemene vi har blitt stilt ovenfor og har måttet løse. Vi vil da konkludere med at dette har vært en svært lærerikt prosess for oss.

Kilder

Ann-Mari Torvatn, (2006)Hovedprosjekter i data/IT – dokumentasjonsstandard, Høgskolen i Oslo, avd. for Ingeniørutdanning.

Burgess. M (2003) Security and online student evaluation with large classes.

Ann-Mari Torvatn, (2001)Kommunikasjon for ingeniører, Tapir Akademisk Forlag.